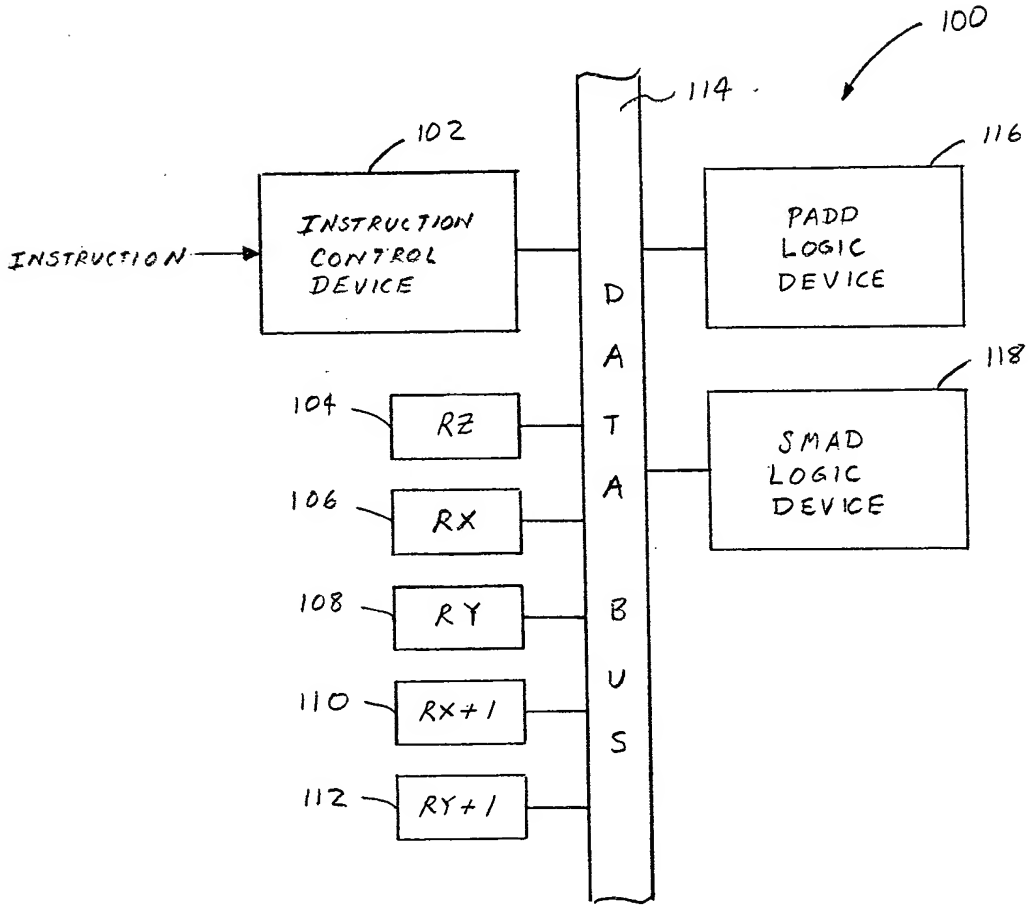
100
114
102
116
INSTRUCTION
INSTRUCTION CONTROL DEVICE
PADD LOGIC DEVICE
DATA BUS
118
104
RZ
SMAD LOGIC DEVICE
106
RX
108
RY
110
RX+1
112
RY+1

FIGURE 1

Instruction Description

200

- **PADD**

Syntax: PADD [-C] [-M] RZ, RX, RY
PADD -I [-C] [-M] RZ, RX, <UI8 : immediate>
PADD -N [-C] [-M] RZ, RX, RY, <UI5 : start>, <UI5 : stop>
PADD -N –I [-C] [-M] RZ, RX, <UI8 : immediate>, <UI5 : start>

RX, RY are the source data registers
RZ is the destination register
<UI8: immediate> specifies the value of an immediate operand
<UI5: start> specifies start of bit field to be modified
<UI5: stop> specifies end of the bit field to be modified
-C indicates addition with carry in
-M indicates addition modulo $2^n - 1$
-N indicates that addition affects only a bit field
-I indicates that second operand is supplied as an immediate value

FIGURE 2A

250

Option Used	Operation
PADD RZ, RX, RY	RZ = RX + RY
PADD –C RZ, RX, RY	RZ = RX + RY + Cin
PADD –I RZ, RX, <UI8: immediate>	RZ = RX + <immediate>
PADD –N RZ, RX, RY, <UI5: start>, <UI5: stop>	RZ = {RX[31:stop], (RX[stop:start] + RY[length] + RX[start: 0]) modulo 2^{length}} Where length = stop – start + 1
PADD –M RZ, RX, RY	RZ = (RX + RY) modulo $2^n - 1$
PADD –N –I RZ, RX, <UI8: immediate>, <UI5: start>	RZ = {(RX[31:start] + immediate[31-start: 0]) modulo $2^{31-start+1}$, RX[start: 0]} In this case, a stop is assumed to be 31.

FIGURE 2B

300

SMAD

Syntax: SMAD [-A] [-M] RZ, RX, RY, <UI2: Length>, <UI2: Num Ops>

RZ is the destination register
RX and RY are source data registers
-A option is used to accumulate results where RZ is used as the accumulator
-M option results in a modulo $2^n - 1$ addition
<UI2: Length> indicates the data widths
- 0: 8 bit operands, where each register is assumed to contain 4 8-bit operands
- 1: 16 bit operands, where each register is assumed to contain 2 16-bit operands
- 2: 32 bit operands
- 3: unused

<UI2: Num Ops> indicates the number of operands to be used in the addition
- 0: 2 source operands RX and RY
- 1: 3 source operands RX, RX+1 and RY
- 2: 3 source operands RX, RY and RY+1
- 3: 4 source operands RX, RY, RX+1 and RY+1

FIGURE 3A

350

Option Used	Operation
SMAD RZ, RX, RY, 2, 0	
SMAD –A RZ, RX, RY, 2, 0	RZ = RZ + RX + RY
SMAD RZ, RX, RY, 2, 3	RZ = RX + RY + (RX+1) + (RY+1)
SMAD RZ, RX, RY, 0, 0	RZ = RX[7:0] + RX[15:8] + RX[23:16] + RX[31:24] + RY[7:0] + RY[15:8] + RY[23:16] + RY[31:24]
SMAD –M RZ, RX, RY, 2, 0	RZ = (RX + RY) modulo 2^n - 1
SMAD –A –M RZ, RX, RY, 2, 0	RZ = (RZ + RX + RY) modulo 2^n - 1

FIGURE 3B

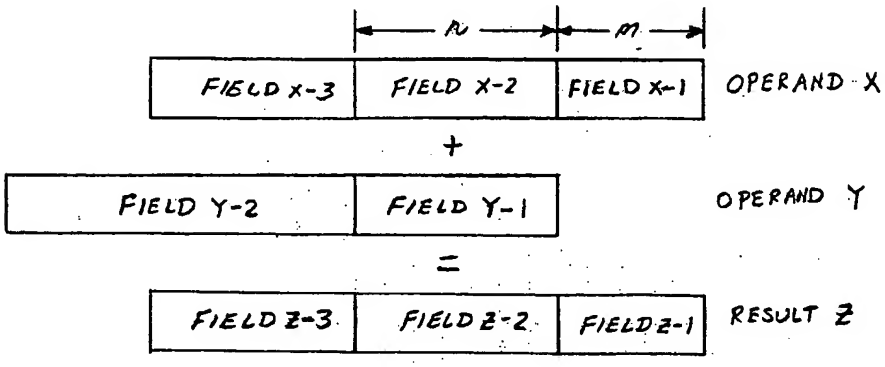

FIGURE 4

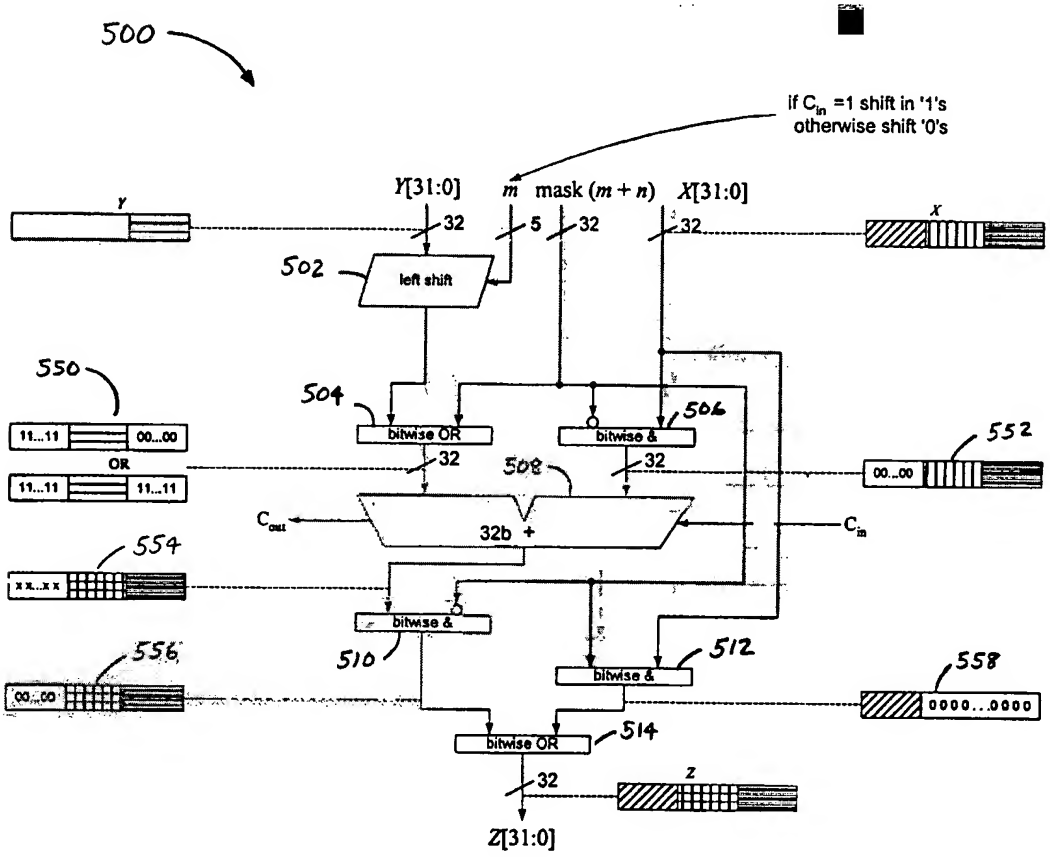

FIGURE 5

m + n	mask (32b)	$\overline{\text{mask (32b)}}$
0: 00000	1111...1110	0000...0001
1: 00001	1111...1100	0000...0011
2: 00010	1111...1000	0000...0111
	.	
	.	.
30: 1_1110	1000...0000	0111...1111
31: 1_1111	0000...0000	1111...1111

Fig. 6

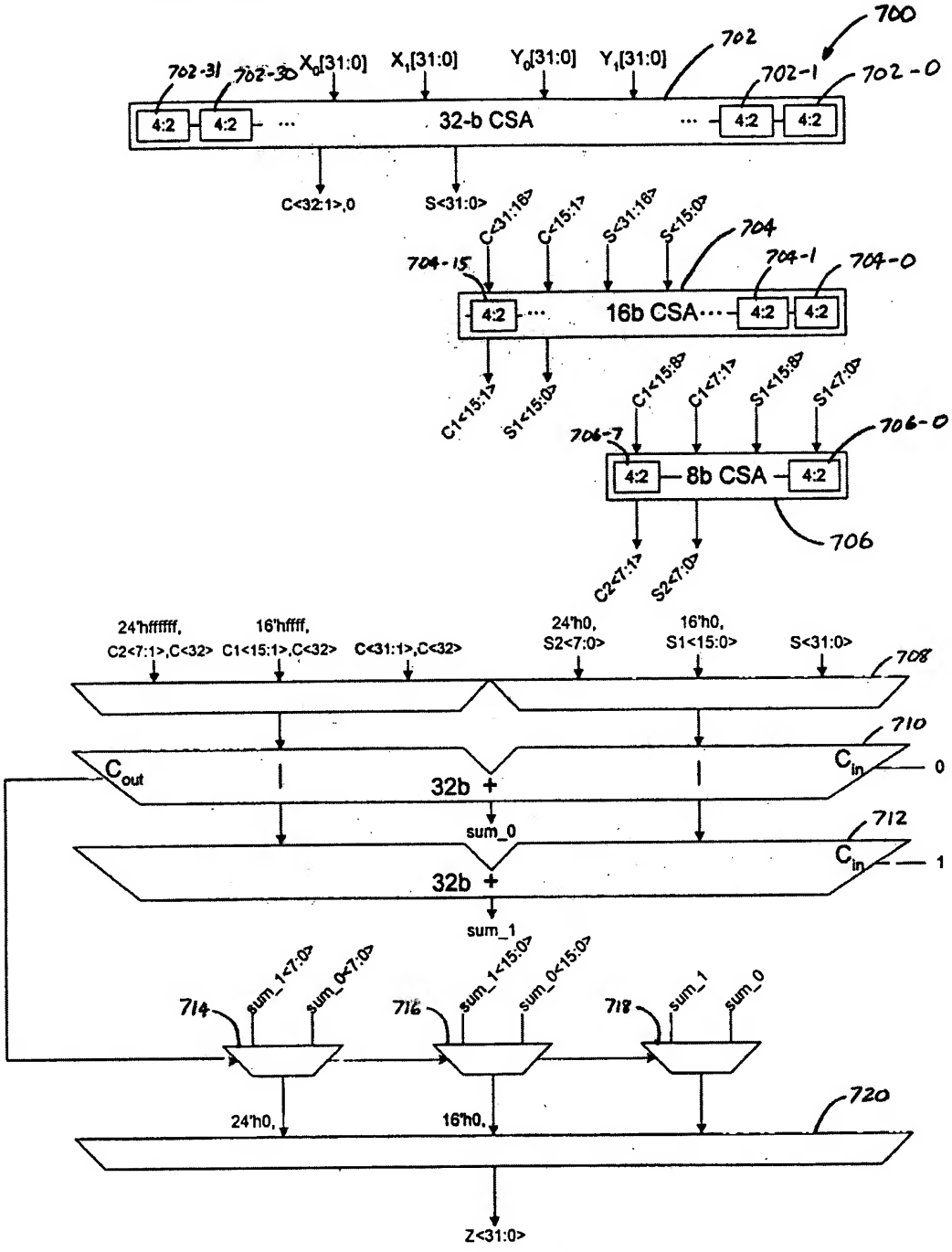
700
702
702-31
702-30
X_0[31:0]
X_1[31:0]
Y_0[31:0]
Y_1[31:0]
702-1
702-0
4:2
32-b CSA
C<32:1>,0
S<31:0>
704
704-15
704-1
704-0
C<31:16>
C<15:1>
S<31:16>
S<15:0>
16b CSA
C1<15:1>
S1<15:0>
706-7
706-0
C1<15:8>
C1<7:1>
S1<15:8>
S1<7:0>
8b CSA
706
C2<7:1>
S2<7:0>
24'hffffff, C2<7:1>,C<32>
16'hffff, C1<15:1>,C<32>
C<31:1>,C<32>
24'h0, S2<7:0>
16'h0, S1<15:0>
S<31:0>
708
710
C_{out}
C_{in}
0
32b +
sum_0
712
1
sum_1
714
sum_1<7:0>
sum_0<7:0>
716
sum_1<15:0>
sum_0<15:0>
718
sum_1
sum_0
24'h0,
16'h0,
720
Z<31:0>

FIGURE 7

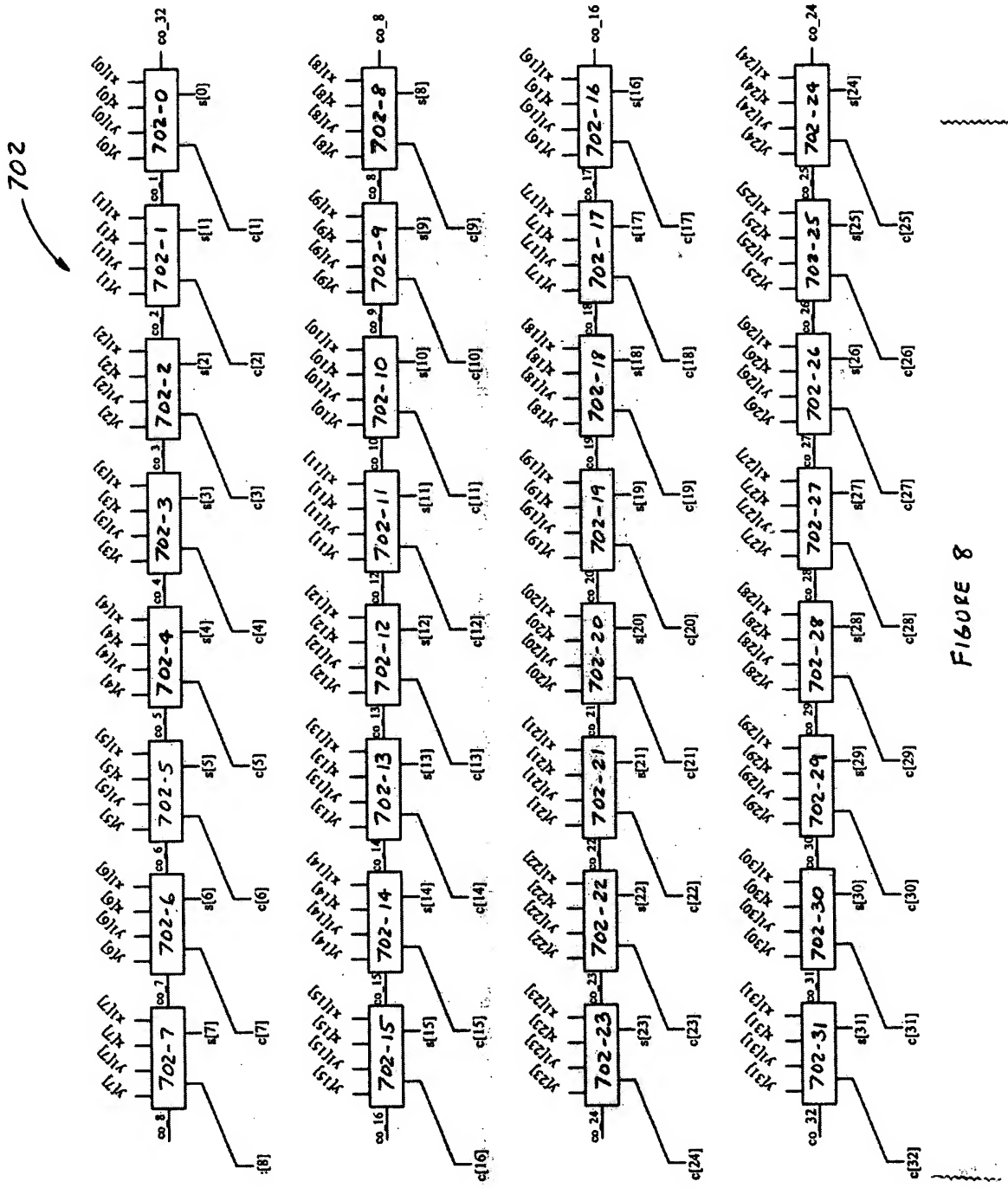
702
co_32
co_8
co_16
co_24
702-0
702-1
702-2
702-3
702-4
702-5
702-6
702-7
702-8
702-9
702-10
702-11
702-12
702-13
702-14
702-15
702-16
702-17
702-18
702-19
702-20
702-21
702-22
702-23
702-24
702-25
702-26
702-27
702-28
702-29
702-30
702-31

FIGURE 8

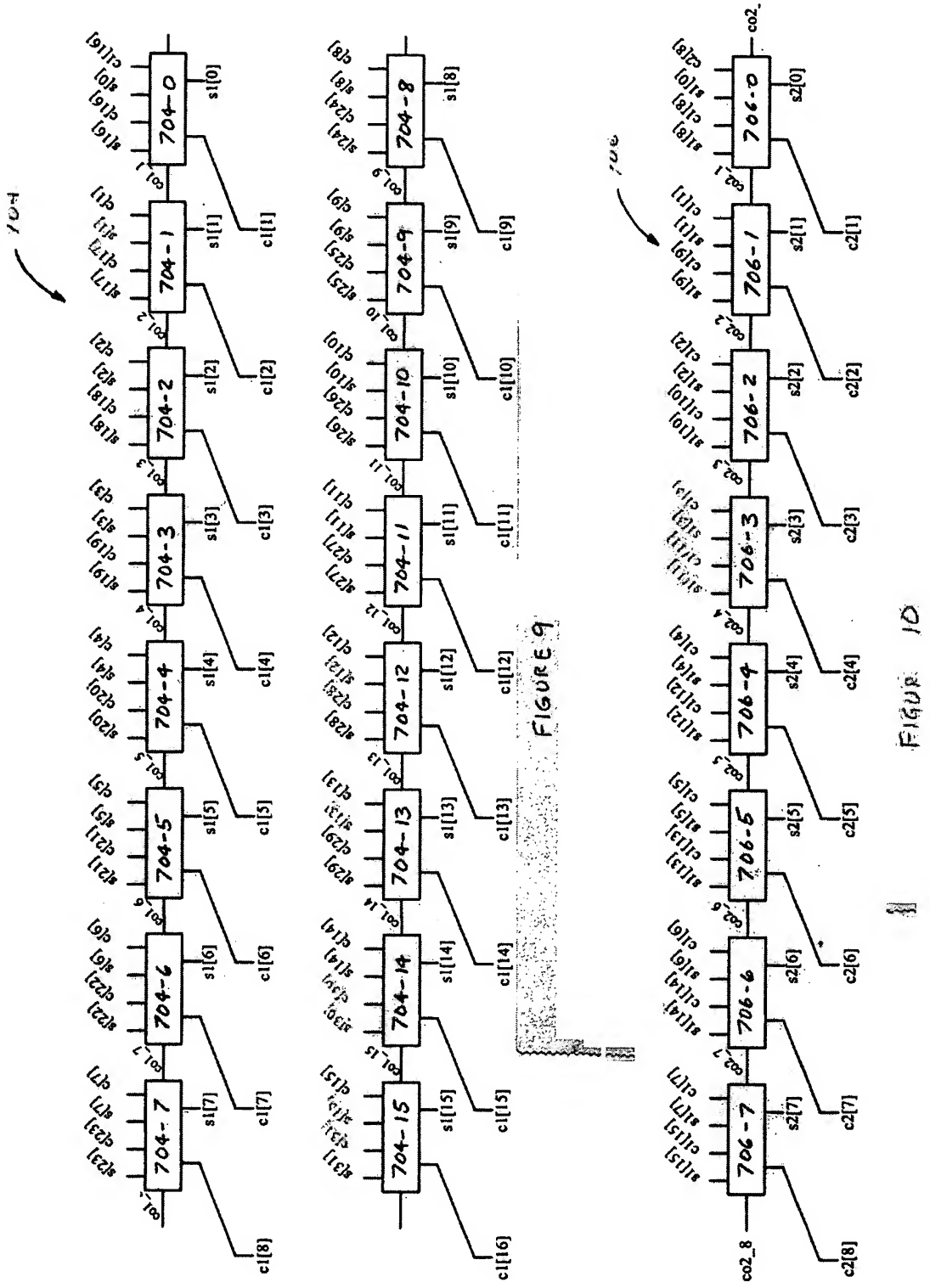

FIGURE 9

FIGURE 10

Carry bits of special consideration in "32-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^n addition, n = ?
c[8]	702-7	704-8	8
co_8	702-7	702-8	8
c[16]	702-15	704-0	8, 16
co_16	702-15	702-16	8, 16
c[24]	702-23	704-8	8
co_24	702-23	702-24	8
c[32]	702-31	708	8, 16, 32
co_32	702-31	702-0	8, 16, 32

FIGURE 11

Carry bits of special consideration in "16-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^n addition, n = ? (n = 32 not applicable)
c1[8]	704-7	706-0	8
co1_8	704-7	704-8	8
c1[16]	704-15	704-0	8, 16
co1_16	704-15	704-0	8, 16

FIGURE 12

Carry bits of special consideration in "8-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^n addition, n = ? (n = 32, 16 not applicable)
c2[8]	706-7	706-0	8
co2_8	706-7	706-0	8

FIGURE 13